Matching Software Practitioner Needs to Researcher Activities

Martin S. Feather
Jet Propulsion Laboratory
California Institute of Technology
Martin.S.Feather@Jpl.Nasa.Gov

Tim Menzies
Lane Dept. of Computer Science
& Electrical Engineering,
West Virginia University
tim@menzies.us

Judith R. Connelly
NASA IV&V Facility
Judith.R.Connelly@nasa.gov

Abstract

We present an approach to matching software practitioners' needs to software researchers' activities. It uses an accepted taxonomical software classification scheme as intermediary, in terms of which practitioners express needs, and researchers express activities. A decision support tool is used to combine these expressions of needs/activities, and to assist in studying the implications of that combined knowledge. This enables identification of fruitful connections between researchers and practitioners, of areas of common interest among researchers, and practitioners, and of "gaps": areas of unfulfilled needs or unmotivated research.

We discuss the software engineering underpinning this approach, illustrating its utility by reporting on experiments with a real-world dataset gathered from researchers and practitioners. We also suggest that this same approach would be applicable to understanding the distribution of interests represented by presenters and attendees of a conference such as APSEC.

1. Introduction

Many organizations look to software engineering research to improve their software products and the software engineering practices by which those products are produced. They wish to reap the benefits that derive from successful technology transfer (the flow of ideas from research to widespread practice) and technology infusion (the adoption and use of research results by specific organizations). However the low rate at which these generally occur has been a continuing concern for decades. [14] reported on a study of impediments to software engineering technology infusion within NASA. This in turn references work from a decade earlier [12]. Many of the observations and insights therein remain valid today. For example, these concerns have recently risen to prominence within the Requirements Engineering community: [6] "…summarises, clarifies and extends..." two conferences' panel discussions on this topic, and [3] address this in their viewpoints article.

One of the impediments to successful infusion is knowing who needs what, and who is working on what. The lack of such knowledge commonly leads to unfulfilled needs, unused research, and unnecessary replication of effort among practitioners who, unbeknownst to each other, share similar problems, and researchers who, unbeknownst to each other, share similar objectives. This problem is exacerbated by the growing number and variety of topics encompassed by software, and the increase in the number of venues (workshops, conferences, journals, web sites etc) in which software research results are reported.

Motivated by these challenges, we have been developing an approach to improve understanding of how software practitioners' needs match to software researchers' activities. The central idea of our approach is to use an accepted taxonomical classification scheme of software as intermediary. Practitioners express their needs in terms of this taxonomy. Researchers express their activities in these same terms. Furthermore, these expressions of activity/interest can be quantitatively weighted to reflect the relative strengths of activity/interest (e.g., a researcher active in several topic areas, but to different degrees) and to reflect the magnitude of the activity/interest (e.g., one research program may be twice the magnitude of another). The gathered data can be used to identify:

- fruitful connections between researchers and practitioners by matching the researchers' combined activities to the practitioners' combined needs,
- areas of overlap among researchers, i.e., opportunities for collaboration and sharing of results, and similar areas of overlap among practitioners, and
- "gaps", areas of needs which are unfulfilled (or only weakly fulfilled) by existing research, and areas of research for which there is little or no demand. This can be useful to inform those planning a research program, to direct new researchers towards areas with high potential, and to redirect ongoing research efforts top better fulfill practitioners needs.

We make inventive use of a decision support tool to represent the information gathered from practitioners and researchers, conduct reasoning across the sum total of that

The 198 leaf nodes, placed side-by-side, of the "software" portion of a taxonomy of computer science

Practitioner with broad range of needs

9 practitioners

Unmet needs!

Overlap – researcher addressing needs shared by several practitioners

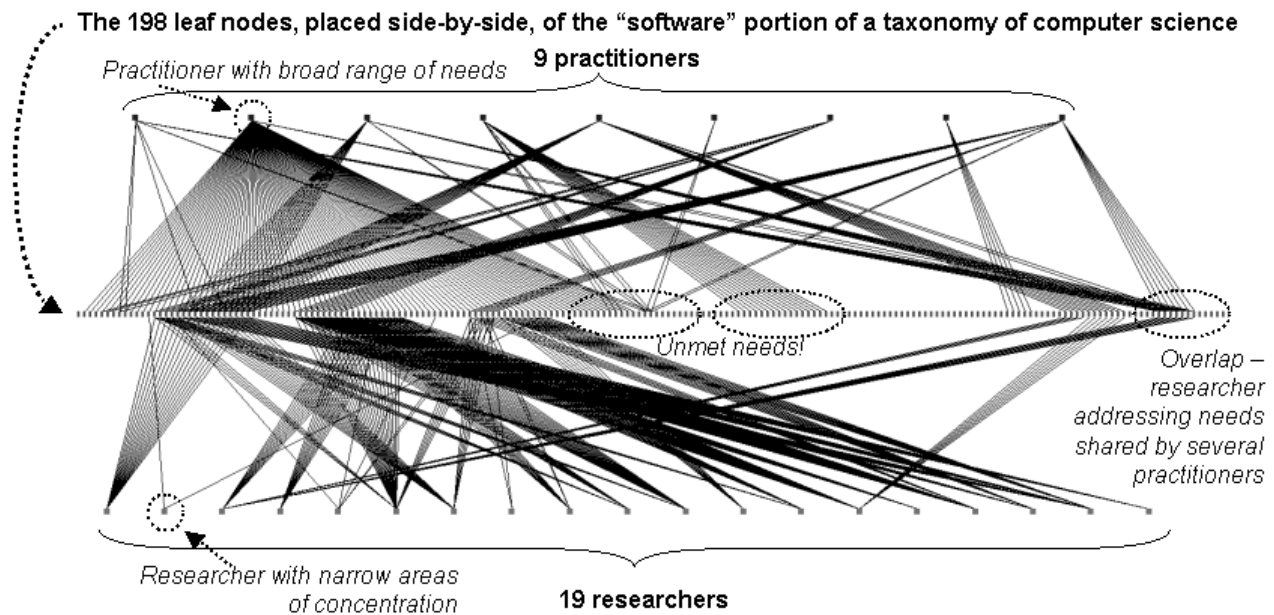*Researcher with narrow areas of concentration*

19 researchers

Figure 1. *Annotated* visualization of 9 practitioners' needs & 19 researchers' activities related via taxonomy

information, and present the results via cogent visualizations.

We have applied this approach to two real-world datasets. One dataset comprises expressions of research activities, gathered from attendees of a focused requirements engineering meeting. Our approach was successful at clearly identifying areas of concentrated, overlapping interests, as well as distinguishing researchers with narrow vs. wide areas of interest. The second dataset comprises expressions of ongoing research from a body of NASA funded software researchers (see [9] & [10] for details of this whole program), and expressions of needs gathered from NASA practitioners in the area of software V&V. Our approach shows the ability to identify the concentrations of research that match practitioner needs, as well as identify areas of need that are receiving little or no attention within from this grouping of research activities. Examples drawn from this second dataset are used as illustrations in this paper (we do not reveal here the identities of the practitioners and researchers).

As motivation to the rest of the paper, Figure 1 shows a visualization of the dataset of V&V practitioners' needs and software assurance researchers' activities, as generated by our decision support tool. We have annotated the diagram to draw attention to the several kinds of important insights that are easy to discern from this form of visualization – *distinctions* between narrowly concentrated vs. broadly dispersed needs/activities; areas of *overlap*, where practitioners' needs and researchers activities coincide; and areas of *unmet* practitioner needs (the right of these two areas contains needs of just one

practitioner, while the left of these areas clearly includes a concentration of several practitioners' needs). It is also evident that all the research activities are in areas in which at least one practitioner has expressed a need.

Present in our data are indications of the *strengths* of these relationships. We will discuss how this additional level of detail is dealt with in our approach, and illustrate other forms of visualization suited to the display of that quantitative data.

A demonstration of this approach is scheduled for presentation at the 11th International Conference on Requirements Engineering, and a paper has been accepted for the 2003 IEEE International Engineering Management Conference covering the utility of this approach from an organizational perspective. The emphasis of this Asia-Pacific Software Engineering Conference paper is twofold:

1. Furnish a description of the software engineering issues underpinning the approach and its realization.

2. Propose the application of this approach at the 10th Asia-Pacific Software Engineering Conference itself. We will to ask the authors of accepted papers to (voluntarily) classify the subject matter of their papers with respect to the software taxonomy, and to ask the attendees of the conference to (voluntarily) classify their areas of software interest with respect to this same taxonomy. At the conference we will demonstrate the approach on this accumulated dataset. We have high expectations that the results will be intriguing and informative to the entire audience.

The remainder of the paper is structured as follows:

Section 2 describes the information elicitation and representation challenges, and the way we have addressed them.

Section 3 explains an analogy between this task of matching practitioners' needs to researchers' activities, and a structurally similar risk management task. This motivates our use of the decision support tool we had developed for risk management applications to this needs/activities matching task.

Section 4 provides the details of our use of the risk management decision support tool in this manner.

Section 5 gives some instances of the kind of visualizations we are able to generate to help gain insights into the needs/activities information.

Section 6 provides related work, and a closing discussion.

2. Information elicitation and representation

Our objective is to be able to compare and reason about multiple practitioners' software needs that research will fulfill, and researchers' activities that are intended to lead to the advances that indeed fulfill those needs. This section considers the challenges of:

- information elicitation - how do we ask practitioners to express their needs, and researchers to describe their activities?, and
- information representation – how do we represent the answers we gather?

We wish to avoid forcing researchers and practitioners to *directly* relate their activities and needs to one another. To do so would require that each researcher understand each and every practitioner problem in order to relate to them, or conversely, require that practitioners understand each and every researcher activity. This "$O(N^2)$" need for understanding (if there are N researchers and N practitioners, it would call for NxN relationships to be considered) is clearly undesirable.

Our solution is the usual software engineering approach, that of interposing an intermediary whose terms are shared by both communities (practitioners and researchers). We use a single taxonomical classification as intermediary between practitioner needs and researcher activities. We rely on several assumptions holding of this classification, namely that it

- already exists,
- is understood by both practitioners and researchers,
- spans the range of concerns involved, and
- goes down to a sufficient level of detail to distinguish among different practitioner needs and different research activities.

We selected the ACM Computing Classification System (1998)[1] [1] as our taxonomical classification scheme. Not only does it meet our assumptions, it is crafted to cover the whole area of computing literature, so indeed spans the range of concerns that arise in software. It goes down to a fairly detailed level, e.g., one of the leaf nodes is "Model checking" (within category D.2.4 Software/Program Verification). We felt that this provides sufficient discriminatory power.

2.1. Information elicitation

We ask practitioners to express their needs for software advances in terms of the ACM taxonomy. Likewise, we ask researchers to express research activities in terms of this same taxonomy. However, the taxonomy is quite large – there are well over 1,000 leaf nodes. Even within the "software" area of the taxonomy, where we wish to concentrate, there are close to 200 leaf nodes. If we were to insist that each expression of interest/activity be stated in terms of leaf nodes, this would make broad ranging needs/activities, which encompass many such leaf nodes, very cumbersome to state.

Our solution is to make use of the tree-structure of the taxonomy for information elicitation purposes: we allow expressions of interest/activity to be stated in terms of nodes at any level of choosing – leaf node or not. Thus if a practitioner has need for advances in the level 3 category D.2.8 Metrics, but does not distinguish between the elements in that category (Complexity measures, Performance measures, Process metrics and Product metrics), then we would allow that practitioner to express interest with respect to that non-leaf-node category in ACM classification tree. Similarly, if a researcher felt that a research activity contributed to the whole of the level 2 category D.4 Operating Systems, then this too could be expressed with respect to that non-leaf-node in the tree.

We also allow for the very likely possibility that an individual might have interests/activities that relate to several nodes, but not to the same extents. For example, a

---

practitioner might see the need for advances to be made in both D.2.8 Metrics, and D.4 Operating Systems, but express (say) twice as much need on the former as the latter. Similarly, a researcher may estimate that a research activity contributes to several areas to differing extents, and so would correspondingly weight expressions of contribution.

Our guidance when eliciting information is as follows: we tell practitioners/researchers to allocate a sum total of 10 "units" among the nodes of the classification scheme. They can subdivide those units any way they like, (including fractional amounts if they wish to make that fine a distinction!). Assigning some number of units to a *non* leaf node is equivalent to subdividing those units equally and allocating those equal amounts to each of that node's children – if this does not match their intent, then they must manually subdivide and allocate in the manner they see fit.

For example, a researcher might state activities as follows:

Node in ACM classification	units
D.2.2 Design Tools and Techniques	2
D.2.4 Software/Program Verification	3
D.2.5 Testing and Debugging	5

Finally, we can also accommodate different total levels of practitioners' needs and different total levels of researchers' activities. For example, a research effort with two full-time researchers will (all other things being equal) be expected to yield twice as many results as one with only one full-time researcher.

2.2. Information percolation

Having gathered the information and attached it to the tree representation of the ACM CCS, we need a way to combine the information. The objective is to gain insight from the combined data into the overall status of research needs vs. activities.

Our first step is to percolate all the weighted expressions of needs and activities down to the leaf nodes of the tree, a simple recursive-descent piece of programming.

For example, the Testing and Debugging node has 10 children in the ACM CCS, so a researcher's expression of 5 units of activity attached to this node would lead to percolation of 0.5 units to each of its children.

The net result of this is a tree whose leaf nodes are labeled by quantified expressions of needs (derived from the expressions of practitioners), and activities (derived from the expressions of researchers). Of course, there can be a mixture of multiple practitioners' needs and/or multiple researchers' activities, with differing quantities of "units", all at a single node.

2.3. Information Combination

Having amassed information at the lowest-level nodes of the taxonomy, we need a way to combine the information at each such node. This poses some challenging questions:

- If several researchers each work in the same area, to what extent do their activities complement or overlap one another?
- Likewise, if several practitioners each have needs in the same area, to what extend do their needs complement or overlap one another?
- How successfully does the combination of researchers' activities in an area fulfill the combination of practitioners' needs in that area?

We do not think that clear-cut answers to these questions can emerge from the (relatively simple) data that we have chosen to gather from both researchers and practitioners. Nevertheless, some plausible trends would seem to hold of this kind of data. For example, the more researchers there are active on a given area, and the greater the intensity of their activities, the more advances will accrue from the sum total of their efforts. However, at least in the short term, some sort of law of diminishing returns would be expected to apply, such that doubling the amount of research in a given area would achieve *less* than double the net advance.

Our approach has been to draw a novel analogy between this task, matching researcher activities to practitioner needs, and an approach to risk-management. The next section considers this analogy in more detail.

3. Adoption of a risk management tool for needs-activities matching

We describe an analogy between the task of matching practitioners' needs to researchers' activities, and a structurally similar risk management task. This motivates our use of the decision support tool we had developed for risk management applications to this needs/activities matching task. A brief description of our risk tool follows.

3.1. Analogy with risk management

The first author has for several years been involved in development and application of an approach to risk management. This approach treats risks as intermediaries between objectives and mitigations. Risks, should they occur, have an adverse impact on objectives, detracting from their attainment. Mitigations, should they be applied, have a reducing effect on risks, by decreasing their likelihood and/or impact.

The idea to apply this risk management approach to the problem of matching practitioners' needs to researchers' activities arose from a similarity we noted between their aims, as follows:

- In risk management, a key aim is to decide upon a cost-effective selection of risk mitigations. The effectiveness of a selection of mitigations is measured as attainment of objectives (risks detract from objectives, and mitigations reduce risks, hence the net effect of a set of mitigations can be calculated in terms of objectives attainment). The cost of a selection of mitigations is the sum of the individual mitigations' costs.
- In matching practitioners' needs to researchers' activities, a key aim could be to decide upon a cost-effective research program to fund. A research program consists of a selection of individual researcher efforts. Its effectiveness is measured as satisfaction of practitioners' needs (impediments to practitioners' needs can be classified by the areas of computer science in which they fall; research within those areas leads to progress towards overcoming those impediments, hence the net effect of a research program can be calculated in terms of fulfillment of practitioners' needs). The cost of a research program is the sum of the individual research efforts' costs.

This analogy is sketched in Figure 2. The key is to consider *lack* of progress within an area of software as equivalent to *risk*. In our risk management methodology, we make the default assumption that risks will occur unless mitigations are applied to reduce them. In our needs/activities matching task, we similarly assume that lack of progress in software areas will persist unless research activity is performed in those areas. Likewise, whether a risk is significant (or not) depends on the extent to which its occurrence would adversely impact objectives, and similarly whether lack of progress in an area of computer science is significant (or not) depends on how much practitioners depend upon progress to be made in that area. In actuality, we may expect that the work of the broader computer science community will eventually lead to progress in all areas of computer science, but a specific institution may wish to direct its own research funding towards solving problems of most importance to that institution as soon as possible.

Further inspiration for this analogy came from our awareness of work at JPL using the risk management approach to assist activity selection across an entire program of NASA Earth Science Missions [13].

3.2. DDP – risk-informed decision support

Our risk management approach, which for historical reasons is called "Defect Detection and Prevention (DDP)", has been developed at JPL and NASA and applied to risk management of spacecraft and spacecraft technologies in their early phases of development. An overview of this work is to be found in [2], and a more extensive description in [4].

The DDP risk management approach rests on the quantitative treatment of risk, in keeping with the vision of using risk as a resource [5]. DDP requires the construction of a risk model in which quantitative estimates are provided of the strength of the impacts between risks and objectives, and the strength of the effects between mitigations and risks.

The purpose of DDP is to aid decision making during the early conceptual stages of development, when detailed designs are lacking, but when the influence of improved decision-making has greatest leverage, because those decisions set the course for the remainder of the development.

In practice DDP has proven successful in this role, leading to early identification of major risks (allowing for them to be addressed sooner and relatively inexpensively rather than later at typically much greater cost), improved selection of mitigations (notably more cost effective selections), and early identification of problematic objectives (those threatened by hard-to-reduce risks).

Custom software has been developed to support the use of DDP, assisting in the elicitation and representation of the risk-related information, calculations over the pooling of that information, and various cogent visualizations for feedback to users. Figure 1 was generated by one of DDP's visualization capabilities. Other examples appear later in this paper.

Recognition of this analogy led us to import our practitioner needs, ACM CCS areas, and researcher activities data into DDP, so as to take advantage of its already-built capabilities.
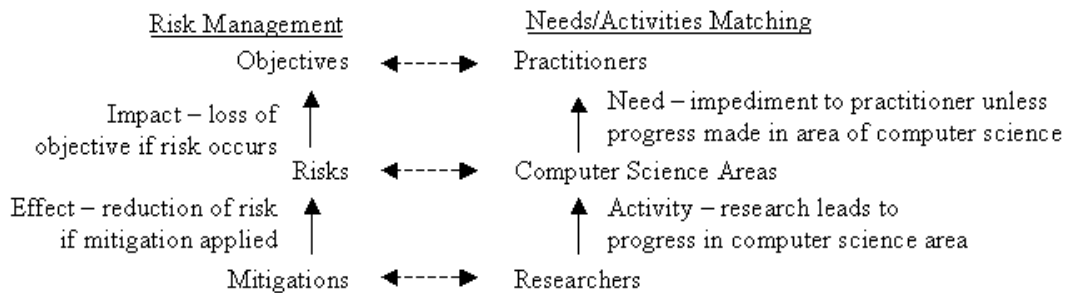
Figure 2. Analogy between Risk Management and Needs/Activities Matching

4. Details of needs/activities represented in the DDP risk-management tool

This section delves into the details of representing the practitioners' needs and researchers' activities as data within the quantitative risk-management DDP tool, and the formulae used by the tool to calculate the net combined effect of that data.

4.1. Importing the practitioner & researcher data into DDP

To import our data into DDP, each practitioner became represented as a separate DDP objective, each researcher became represented as a separate DDP mitigation, and the entire ACM Computing Classification System tree became represented as a DDP risk tree (the DDP tool supports tree-structured data). Since the vase majority of our data ended up concentrated within the "Software" subtree of the ACM CSS, we focused on just that portion.

Each practitioner's expression of need was percolated down to the leaf level as described earlier, by a simple piece of recursive descent code added to DDP for this purpose. Each of the resulting weighted expressions of need was used to link the practitioner to the leaf node in the ACM CCS, scoring that link with the weight computed in the percolation process.

Similarly, each researcher's expression of activity was percolated down to the leaf level. Each of the resulting weighted expressions of activity was used to link the researcher to the leaf node in the ACM CCS, scoring that link with the weight computed in the percolation process.
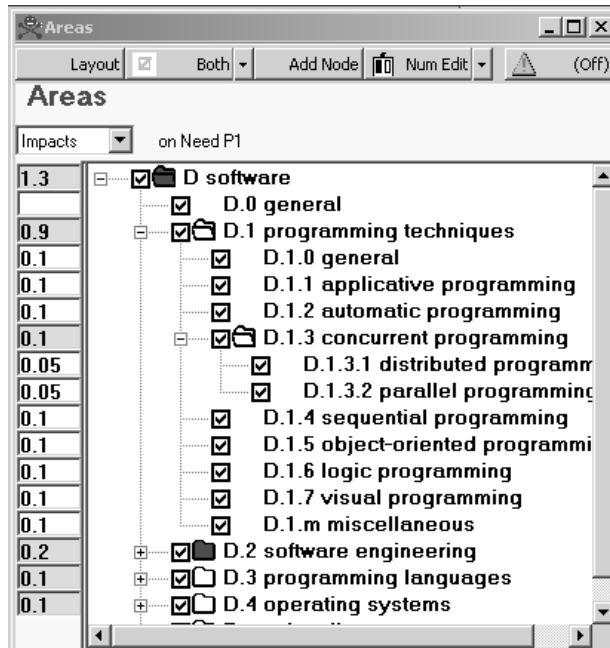

Figure 3. ACM CCS viewed within DDP

We gathered these expressions of needs and activities by emailing people the URL to the ACM CCS, and asking them to email back to us a list of their chosen nodes of the CCS, each weighted by the strength of their need/activity in the area represented by that node. As discussed earlier, their expressions could be stated in terms of leaf and/or non-leaf nodes of the CCS. We then entered their data into our risk management tool ourselves (thus avoiding requiring each person to download, install and run the tool themselves).

The DDP tool supports data entry, so our task was straightforward. If someone who knows how to use the tool is on hand, then practitioners and researchers can view the ACM CCS through the tool's GUI, and be guided on how to enter their expressions directly. Figure 3 shows a partial screenshot of the software portion of the ACM CCS as viewed within the DDP tool. Nodes of the tree (seen occupying most of the acreenshot) can be expanded and collapsed to reveal more or less levels of detail (e.g., the area D.1 programming techniques is shown fully expanded, while the area D.2 software engineering is shown collapsed). The list of slots down the left are for view and entry of values, in this case the values representing practitioner P1's "needs" in each of the computer science areas.

4.2. Quantitative treatment of the practitioners' needs data

The risk analogy also gives a suggestion as to how to quantitatively treat the combined needs and activities data.

In risk-centric DDP applications, the extent to which an objective is *unmet* is determined by the sum of risks impacting that objective. By analogy, the extent to which a practitioner's needs are unmet is determined by the sum of the unachieved advances impeding the computer science area needs of that practitioner.

More precisely, in risk-centric DDP applications:
- an objective may be impacted by risks;
- each such impact is scored by the proportion of the objective that would be lost if that risk were to occur. For risk R and objective O we will write this as: Impact(R, O)
- each risk's likelihood of occurrence is calculated from its a-priori likelihood and the risk-reducing effects of selected mitigations. For risk R we will write this as Likelihood(R)

DDP automatically calculates an objective O's "at-risk" measure as:

$$\Sigma\ (R \in \text{Risks}): \text{Impact}(R,O) * \text{Likelihood}(R)$$

and similarly calculates an objective O's "a-priori risk" measure (how much total loss of objectives the *un*mitigated risk causes) as:

$$\Sigma\ (R \in \text{Risks}): \text{Impact}(R,O) * \text{A-PrioriLikelihood}(R)$$

So for our needs-activities task:
- a practitioner may be impeded by need for progress in computer science areas;
- each such impedance is scored by the proportion of the practitioner's weights given to that area. For example, if a practitioner had identified three areas of need, and weighted them 15, 15 and 30, then the first and second of these would each score 0.25 (15 / (15 + 15 + 30)), and the third would score 0.5 (30 / (15 + 15 + 30)). For practitioner P and area A we will write this as: Need(P,A)
- each area has some potential for progress; the expected contributions of research activities in that area contribute to such progress. For an area A we will write this as Progress(A)
- areas don't have an "a priori" likelihood, so where this would occur in a DDP formula, it is replaced by the value 1.

Hence a practitioner P's total amount of "need" for research is:
$$\Sigma \ (A \in \text{Areas}): \text{Need}(P,A)$$
Similarly, a practitioner P's total amount of "*unmet*-need" for research (i.e., taking into account progress expected from research) is:
$$\Sigma \ (A \in \text{Areas}): \text{Need}(P,A) * (1 - \text{Progress}(A))$$

As well as calculating the "at-risk" measure for objectives, risk-centric DDP also calculates the "total impact" measure for risks. For risk R, the formula is:
$$\Sigma \ (O \in \text{Objectives}) \ \text{Impact}(R,O) * \text{Likelihood}(R)$$
In practice, some objectives are more important than others. This is captured by giving each objective a "Weight", reflecting its relative importance. Taking this into account, the above formula becomes:
$$\Sigma \ (O \in \text{Objectives}):$$
$$\text{Weight}(O) * \text{Impact}(R,O) * \text{Likelihood}(R)$$
Similarly, in our needs-activities matching tasks, some practitioners are more important than others (for example, they may be responsible for the V&V of a larger program area). If we use "Weight" to capture relative importance in the same manner, the "total unmet need" measure for area A is:
$$\Sigma \ (P \in \text{Practitioners}):$$
$$\text{Weight}(P) * \text{Need}(P,A) * (1 - \text{Progress}(A))$$
If there were no research activities taking place, then Progress(A) would equal zero for each area A, and the formula would simplify to
$$\Sigma \ (P \in \text{Practitioners}): \text{Weight}(P) * \text{Need}(P,A)$$
Intuitively, this gives a quantitative measure of the total practitioner need for advances in each of the software areas.

4.3. Quantitative treatment of the researchers' activities data

We now consider how to interpret the researchers' activities data.

In risk-centric DDP applications, mitigations reduce risks, and so lead to greater attainment of objectives. By analogy, researchers' activities contribute advances to areas of computer science, and so lead to meeting more of the needs of practitioners.

More precisely, in risk-centric DDP applications:
- a risk may be effected by mitigations. (Usually the effect is a reduction of either the risk likelihood or the risk severity; DDP also allows for the case that a mitigation makes certain risks worse. For the analogy used here, only risk reduction is relevant);
- each such effect is scored by the proportion by which the risk would be reduced if that mitigation were applied. For mitigation M and risk R we will write this as: Effect(M,R)
- each risk's likelihood of occurrence, Likelihood(R), is calculated from its a-priori likelihood and the effects of applied mitigations, thus:

$$\text{A-Priori}(R) * \Pi \ (M \in \text{Mitigations}): (1 - \text{Effect}(M,R))$$

Intuitively, mitigations act like "filters", each filtering out some proportion of the incoming risks, with multiple filters arranged in series. For example, if one mitigation's effect is 0.9, it filters out 90% of the incoming risks, leaving 10% remaining. A second filter whose effect is 0.5 would filter out 50% of the risks that got through the first filter, leaving just 5% of the original risks remaining.

So for our needs-activities task:
- a researcher's activities may contribute research advances to areas of computer science;
- each such contribution is scored by the magnitude of the advances in that area. For researcher R and area A, write this as Contribution(R,A)
- in an area A the combination of multiple researchers' contributions within that area leads to a measure of Progress(A) =

$$1 - (\ \Pi \ (R \in \text{Researchers}): (1 - \text{Contribution}(R,A)) \ )$$

For example, if there were no researchers active in a given area, its Progress measure would be 0; if there was just one researcher active with Contribution=0.8, then its Progress measure would be $(1 - (1-0.8)) = 0.8$; if there were two researchers with Contributions 0.8 and 0.5, then its Progress measure would be $(1 - (1-0.8)*(1-0.5)) = 0.9$ This formula captures the intuitive notion of some overlap among the researchers activities within a given area.

There is an underlying decision we must make about the effectiveness of research activities. For example, in the extreme case of a researcher whose entire research effort is devoted to a single leaf node area in the ACM CCS,
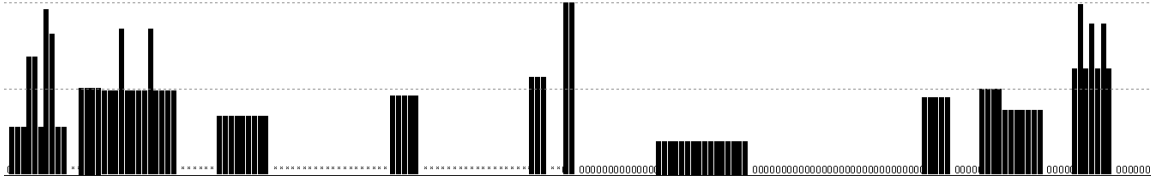
Figure 4. Bar chart of magnitude of total practitioner need for each software area (heights in log scale)

how completely does that fulfill need for research progress in that area? In terms of our formula, this is equivalent to asking what would be the value for Contribution(R,A). Since we do not know the answer to this per se, we make this a parameter of the DDP model, and experiment with different settings for its value. When assigned a high value, this models the assumption that it takes relatively little research activity to achieve progress, whereas when assigned a low value, this models the assumption that it takes a lot of research activity to achieve the same amount of progress.

5. Utilizing visualizations for decision support

One of the benefits of using the DDP risk management tool is the pre-built visualizations it offers for scrutinizing its risk information in aid of decision making. Having encoded the practitioners' needs information and researchers' activities information in DDP, we are able to make use of these visualization capabilities.

5.1. Visualizations of practitioners' needs information

Figure 4 shows a bar chart of the magnitudes of the total practitioner need for advances in each of the software areas, assuming each practitioner is equally important. There is one bar per software area, the height of which indicates the logarithm of the magnitude of the need. Because of the log scale, the bottom of the chart corresponds to a small magnitude, not to zero. Areas whose magnitude of need is smaller than this are labeled

with a tiny "*", while areas whose magnitude of need is zero is labeled with a tiny "0". For example, some closely spaced clusters of "0's" are discernable at the right hand side of the chart.

The areas to which the bars correspond are in the same sequence as the areas plotted across the middle of Figure 1. This correspondence is evident if we juxtapose the top half of Figure 1 (the lines linking practitioners to the areas they need) and a vertically inverted version of Figure 4, to form Figure 5. From this juxtaposition it is evident that concentrations of practitioner needs lead to higher magnitude quantitative measures of need, just as we would expect to see.

It might appear that the bar chart conveys little additional information beyond that in the link chart. The link chart, however, is less adept at conveying the results of variations in strengths of relationships, and variations in the relative weights of practitioners. For simplicity, this chart was generated assuming each practitioner to be of equal importance. When this is not the case, the quantitative consequences are apparent only through the bar chart visualization.

One of the useful DDP options is to sort bars in such charts into descending order, from which we can see the areas most needing research advances. In the interests of space, this easy-to-imagine view is omitted from this paper.

Overall, this kind of bar chart presentation of practitioners' needs information makes evident the uneven concentrations of needs in certain areas. This would be useful information to guide a new research program, or to guide a search for existing research results.
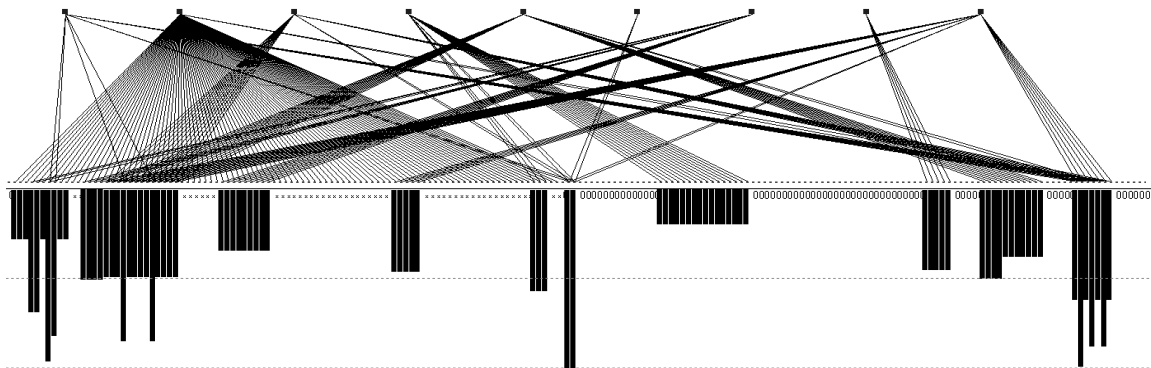

Figure 5. Juxtaposition of practitioner-to-areas links, and magnitudes of total practitioner need per area
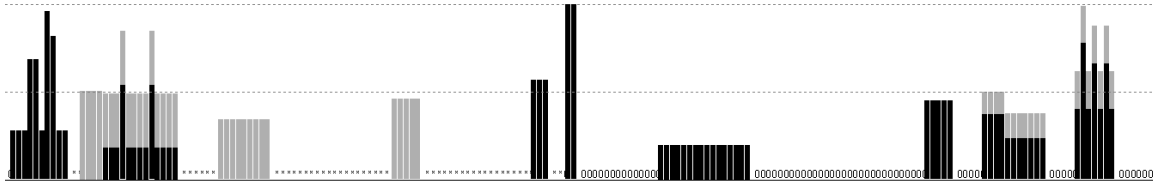
Figure 6. Bar chart of magnitude of total (black) and met (gray) practitioner need for each software area

5.2. Visualizations of researchers' activities information

An obvious area of interest is in ascertaining to what extent the researchers' activities will lead to fulfillment of the practitioners' needs. This is the "unmet-need" calculation described in section 4.2.

Figure 6 shows the results of this calculation, plotted in the same bar-chart style as before. In this chart gray bars show where levels of practitioner need started (i.e., the bar heights of Figure 4), and black bars show remaining levels of need – the effects of the researchers' activities have been to reduce these remaining levels of need, of course.

From this chart it is immediately evident that there are several areas of significant need on which no researchers are working (or their contribution is so minor as to fail to be discernable at this scale). For example, the two tallest bars (so close that they look almost like one) in the center of the chart are the tallest in the whole chart, and yet there is evidently no research being done in these areas.

DDP offers several sorting options (sort by level of remaining need, sort by amount of need fulfilled by research, etc). Again, these easy-to-picture options are not shown here in the interests of space.

5.2.1 Gaining insights from other DDP supported visualizations

Figure 7 shows a very different kind of visualization available in DDP. This takes the overall form of lists-of-lists. In this instance, the major list (the rows) comprises the areas of need of one of the practitioners (the leftmost practitioner in the row at the top of Figure 1). That practitioner's expressions of needs have percolated down to 13 distinct leaf nodes in the ACM CSS, each of which is represented as a separately listed row. The large rectangles down the left side correspond to these areas – they are labeled with the area (e.g., "D.1.5"), and their widths are proportional to the sum total outstanding need for research in their areas. To the right of each large rectangle is a (possibly empty) list of smaller rectangles, each corresponding to a researcher active in that area. Of course, the same researcher may occur in several rows, contributing to each of the areas to which those rows correspond. The width of the researcher's box indicates the strength of that researcher's contribution to that area. Finally, the checkboxes indicate and control (toggle their

checked/unchecked status with mouse clicks) whether or not the researcher's activities are assumed to be taking place when computing the amounts of fulfilled, and remaining unfulfilled, needs.

This visualization makes it easy to discover which researchers are working on which of the areas relevant to a practitioner. For example, the third and fourth of the areas (labeled D 2.2.4 and D 2.2.9) each have the same three researchers active in those two areas (researchers labeled R1, R7 and R18). R1's boxed are wider than those of R7 and R18, indicating R1 has more activity in those areas. For the first, second, fifth and sixth areas there are evidently no researchers active, while for the bottom seven areas, there is just researcher R3 active. In the interests of space, we picked one of the practitioners with relatively few areas of need; this same kind of visualization scales well to other practitioners with greater number of needs (in our dataset, the second of the practitioners has 36 areas of needs).

There are analogous visualizations in which areas are listed alongside the researchers fulfilling them, practitioners are listed alongside areas they need, etc.
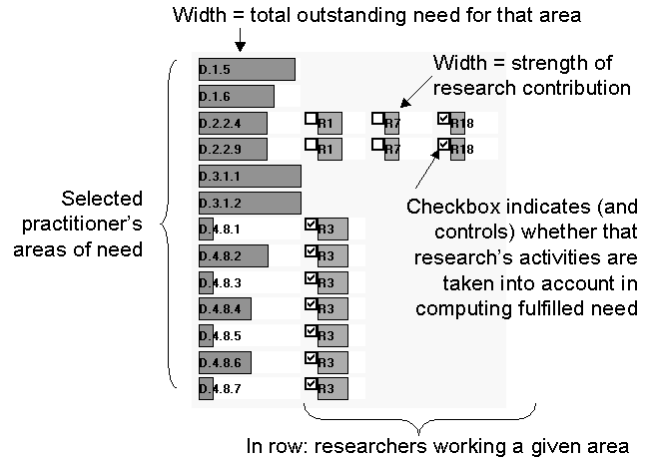

Figure 7. Needs-to-researchers lists

6. Discussion

6.1. Related work

Our objective of matching practitioners' needs to researchers' activities is closely related to the classic requirements analysis problem of matching features of a

to-be-developed product to customer needs. Representative work in this area includes:

Karlsson & Ryan's study of selection of requirements for software system developments [7]. Their approach yields a 2-dimensional "cost-value" diagram in which each requirement is plotted as a point located according to its customer value in one dimension, and cost of implementation in the other dimension.

Kulik & Macdonald's approach to classifying project requirements into the major categories of "Add Value", "Must Do", "Nice to Have" and "Defer" [8]. Their method combines results of Quality Function Deployment and Kano Analysis into a 2-dimensional "needs-opportunity" diagram in which each requirement is plotted as a circle centered at the point located according to that requirement's degree of customer need in one dimension, and proportion of customers who have that need in the other dimension; radius of the circle indicates a measure of the Return On Investment that requirement represents.

We might attempt these approaches in our problem area by plotting computer science areas as requirements (valued in terms of practitioners' needs, and costed in terms of researchers' activities). However, we deal with a much larger number of items (almost two hundred leaf nodes in just the "Software" category of the ACM Computing Classification System) compared to the 20 or so requirements on which these authors illustrate their work. We also seem faced with a more open-ended decision space than the equivalent of seeking the optimum set of requirements for a given cost level. For example, we wish to use the information we have gathered to give insights into future areas where research would be beneficial, as well as understand how the identified set of activities meets the practitioners' needs.

Note that asking a practitioner to rank the relevance of each researcher's set of activities is not a viable alternative. It assumes too much knowledge by the practitioner of the research activities, and requires continued update by the practitioner as more researchers are added. It also precludes recognition of the situation that the union of several researchers' activities together meets the practitioner's needs. By a similar argument, researchers cannot rank their relevance to each practitioner's problem. The use of the intermediary taxonomy, familiar to both sides, is key. As mentioned before, we got inspiration for this from JPLer David Tralli's use of DDP to assist activity selection across an entire program of NASA Earth Science Missions [13].

6.2. Conference application

We propose the application of this approach at the 10th Asia-Pacific Software Engineering Conference itself. We will to ask the authors of accepted papers to (voluntarily) classify the subject matter of their papers with respect to the software taxonomy, and to ask the attendees of the conference to (voluntarily) classify their areas of software interest with respect to this same taxonomy. At the conference we will demonstrate the approach on this accumulated dataset, using the DDP tool's visualizations to present and explore the accumulated data. We have high expectations that the results will be intriguing and informative to the entire audience.

6.3. Conclusions

The overall aim of this work is to match practitioners' needs to researchers' activities so as to gain insights into the status of entire research programs. These insights should benefit organizations that fund, direct and/or utilize research, researchers who wish to know areas are in need of research and by whom, and practitioners who wish to know what research activities are taking place and who is performing them.

The two key steps of our approach are:
1. Employing a taxonomical classification scheme as intermediary between expressions of need and expressions of activity. This was key to successfully eliciting from practitioners expressions of needs, and from researchers expressions of activities, and thereafter combining them.
2. Inventive use of a risk-centric decision-support tool, which both
   - suggests a useful analogy in which lack of progress in a given area is a "risk" that adversely impacts attaining practitioners' needs, and which can be mitigated through the contributions of researchers' activities, and
   - provides the mechanical support needed to handle the volume of information. DDP's mechanisms for information visualization have proven useful for presenting the information in such a way that insights can be made despite the volume of information.

We were able to use the DDP tool's capabilities for calculation and visualization as is, with the only additional work needed being a small amount of programming to import the data.

One of the assumptions buried within our approach is the definition of how to calculate the contribution of a set of research activities towards meeting a need for advances within a given area. The formulae we used were motivated by the analogy with risk mitigation, but this is not the only possible way of deciding upon this form of calculation. Our feeling is that this problem falls into the category that Ritell termed "*wicked problems*" [11]. Wicked problems have many features, the most important being that no objective measure of success exists. Designing solutions for wicked problems cannot aim to produce some perfectly correct answer since no such definition of *correct* exists. Our approach will be to

experiment with several variations of data combination, and find which of the conclusions we extract from the resulting data remain stable across many/all of those variations.

6.4. Status and future work

The status of our work is that the data gathered from 9 NASA assurance researchers and 19 V&V practitioners has been successfully imported into DDP in the manner described and illustrated herein. A different set of data, comprising expressions of research activities gathered from attendees of a special purpose requirements engineering meeting, has also been entered and studied. On this latter dataset (not shown in this paper) our approach is able to clearly identify overlapping interests, as well as distinguish researchers with narrow vs. wide areas of interest.

Future work will be to inject this capability into the research planning and management processes. The hope is that armed with the kind of information that this approach reveals, research program managers will be better able to match their programs to the emerging needs of long-lived projects. The extension of this approach to study *trends* of research and application is also an area of interest.

Our approach relies upon a single taxonomy that classifies the areas, sub-areas, etc. of a domain. Taxonomies have limitations, e.g., they force an item to be classified in a single location (which can be problematic). Furthermore, our scheme uses an evaluation mechanism that is based on a weighted sum of the leaf items of a taxonomy. Current interest in ontologies is motivated by the recognition that ontologies offer more support for inference than do taxonomies. Thus an avenue for exploration would be to extend our (relatively simple) approach to the richer opportunities that ontologies offer.

7. Acknowledgements

8. References

[1] ACM Computing Classification System [1998 Version] http://www.acm.org/class/1998/

[2] S.L. Cornford, M.S. Feather & K.A. Hicks. "DDP – a tool for life-cycle risk management", Proceedings *IEEE Aerospace Conference, 2001*, Big Sky, MT Volume 1, pp. 1/441-1/451.

[3] A.M. Davis & A.M. Hickey. "Requirements Researchers: Do We Practice What We Preach?", *Requirements Engineering* 7(2), June 2002, pp. 107-111.

[4] Feather, M.S. & Cornford, S.L.. "Quantitative risk-based requirements reasoning", to appear in *Requirements Engineering* (Springer), published online 25 February 2003, DOI 10.1007/s00766-002-0160-y. Available from: http://eis.jpl.nasa.gov/~mfeather/AvailablePublications/

[5] M.A. Greenfield "Risk Management: 'Risk As A Resource'" http://www.hq.nasa.gov/office/codeq/risk/

[6] H. Kaindl, S. Brinkkemper, J.A. Bubenko Jr, B. Farbey, S.J. Greenspan, C.L. Heitmeyer, J.C.S. do P. Leite, N.R. Mead, J. Mylopoulos & J. Siddiqi. "Requirements Engineering and Technology Transfer: Obstacles, Incentives and Improvement Agenda", *Requirements Engineering* 7(3), Oct. 2001, pp. 113-123.

[7] J. Karlsson & K. Ryan. "A Cost-Value Approach for Prioritizing Requirements", *IEEE Software*, Sept./Oct. 1997, pp 67-74.

[8] P. Kulik & D. Macdonald. "Bridging Gaps in Customer Requirements", *5th European Project Management Conference, PMI Europe 2002*, Cannes, France, Jun 2002.

[9] *The 1st Annual Office of Safety and Mission Assurance Software Assurance Symposium*, Sept. 2001. http://www.ivv.nasa.gov/business/research/osmasas/conclusion.htm

[10] *The 2nd Annual Office of Safety and Mission Assurance Software Assurance Symposium*, Sept. 2002. http://www.ivv.nasa.gov/business/research/osmasas/conclusion2002.htm

[11] H.W.J. Rittel. "Second generation design methods", *Design Methods Group 5th Anniversary Report*: DMG Occasional Paper, 1, 5-10. Also in N. Cross (Ed.) *Developments in design methodology*, (pp. 317-327). Chichester: Wiley & Sons, 1984

[12] E. Rogers. *Diffusion of Innovation*. The Free Press, New York, 1983.

[13] D.M.Tralli. Programmatic Risk Balancing. *IEEE Aerospace Conference*, Big Sky MT, 2003.

[14] M.V. Zelkowitz. "Software engineering technology infusion within NASA", *IEEE Transactions on Engineering Management*, 43(3), Aug. 1996, pp. 250-261.